

Attorney Docket No.: 00CON102P

REMARKS

Prior to the present amendment and response, claims 1-9, 11-15, and 21-28 were pending in the present application. By the present amendment and response, claims 1, 9, and 21 have been amended; thus, claims 1-9, 11-15, and 21-28 remain in the present application. In view of the above amendments and the following remarks, allowance of outstanding claims 1-9, 11-15, and 21-28 is respectfully requested.

The Examiner has rejected claims 1-9 and 11-28 under 35 USC §102(e) as being anticipated by U.S. Patent Number 6,615,338 to Tremblay, et al. (hereinafter "Tremblay"). For the reasons discussed below, Applicants respectfully submit that the present invention, as defined by independent claims 1, 9, and 21, is patentably distinguishable over Tremblay.

Various embodiments according to the present invention relate to an improved performance VLIW processor. Some previous attempts at VLIW processors, such as Tremblay, result in an advantage in parallel processing of a number of instructions. Nevertheless, these VLIW processors exhibit unnecessary power consumption. On page 7, paragraphs 17 and 18 of the present final rejection, the Examiner has pointed to two paragraphs in Tremblay. The first paragraph pointed to by the Examiner appears on column 7, lines 30-38 in the detailed description section of Tremblay, and is quoted below in its entirety:

"The pipeline control unit 226 is connected between the instruction buffer 214 and the functional units and schedules the transfer of instructions to the functional units. The pipeline control unit 226 also receives status signals from the functional units and the load/store unit 218 and uses the

status signals to perform several control functions. The pipeline control unit 226 maintains a scoreboard, generates stalls and bypass controls. The pipeline control unit 226 also generates traps and maintains special registers.” Tremblay, column 7, lines 30-38.

The second paragraph pointed to by the Examiner appears on column 1, line 64 to column 2, line 5 in the background section of Tremblay, and is quoted below in its entirety:

“VLIW processors package multiple operations into one very long instruction, the multiple operations being determined by sub-instructions that are applied to the independent functional units. An instruction has a set of fields corresponding to each functional unit. Typical bit lengths of a subinstruction commonly range from 16 to 64 bits per functional unit to produce an instruction length often in a range from 64 to 512 bits for VLIW groups from four to eight subinstructions.” Tremblay, column 1, line 64 to column 2, line 5 (emphasis added).

Applicant respectfully points out that not only these two paragraphs are unrelated, indeed one belongs to the detailed description of Tremblay and the other belongs to the background section of Tremblay, but also these paragraphs are taken out of context. In any event, even if it were successfully argued that combining these two unrelated paragraphs does not amount to impermissible hindsight reconstruction, any such combination falls far short of what the present invention teaches.

To further clarify, the invention teaches a scheme of forced division of a VLIW packet into issue groups no greater than 64 bits. This is disclosed, for example, in Page 20, lines 11-18 of the present application:

“In the present embodiment of the invention, the assembly code written for the VLIW processor consists of VLIW packets with one issue group having 64 bits and the other issue group having 48 bits. Thus, if a

particular VLIW packet contains only one issue group, the VLIW packet is divided up into two issue group, with one issue group being 64 bits and the other being 48 bits. Moreover, the VLIW packets are not permitted to have three or more issue groups. Thus, in the present example, all VLIW packets processed by the invention's VLIW processor 300 would contain exactly two issue groups, one issue group being 64 bits and the other issue group being 48 bits." Page 20, lines 11-18 of the present application (emphasis added).

Tremblay is not directed to, nor does it even suggest, the forced limitation of each issue group to any number of bits or, more particularly, to 64 bits. Indeed, the portion of Tremblay relied upon by the Examiner states that each subinstruction can be between 16 and 64 bits long, while each issue group in the VLIW packet would consist of four to eight instructions, thus ranging between 64 and 512 bits: "Typical bit lengths of a subinstruction commonly range from 16 to 64 bits per functional unit to produce an instruction length often in a range from 64 to 512 bits for VLIW groups from four to eight subinstructions." Tremblay, column 2, lines 2-5. In other words, far from limiting the number of bits in each issue group to 64 bits, Tremblay indicates that the number of bits in each of its issue groups can be a wide range, starting from 64 bits (and up to 512 bits). However, the present invention is directed to a two-thread processor, with each thread being required to process an issue group less than or equal to 64 bits.

Applicant refers the Examiner to the advantages of the present invention flowing from the invention's scheme of forcing a limit, i.e. the claimed limit of 64 bits, on each issue group being processed in a respective thread. One such advantage is to reduce the unnecessary power consumption resulting from conventional approaches. One reason for

such unnecessary power consumption in conventional processors is illustrated with the aid of an example provided by reference to Figure 2 of the present application:

“After exemplary VLIW packet 200 is fetched from a cache or an external memory, the four instructions in VLIW packet 200 must be forwarded to appropriate execution units for execution. To account for the possibility that all of the instructions in a given VLIW packet may belong to a single issue group, the instruction bus coupled to the execution units of the VLIW processor must be 112 bits wide to carry all four instructions in the VLIW packet at the same time. However, as illustrated in the present example, the first issue group consists of merely two long instructions requiring an instruction bus that is only 64 bits wide while the second issue group consists of merely one long instruction and one short instruction requiring an instruction bus that is only 48 bits wide. Thus, in the case of exemplary VLIW packet 200, an instruction bus that is 64 bits wide is all that is needed to handle the processing of both the first and second issue groups in the VLIW packet. As such, a 112-bit wide instruction bus would result in an unnecessary power consumption associated with 48 bus lines that are not needed in the processing of exemplary VLIW packet 200. Further, an instruction bus which is 112 bits wide requires considerably greater chip area as compared with an instruction bus which is only 64 bits wide.” See page 4, line 20 to page 5, line 12 of the present application.

As such, conventional VLIW processors have an architectural limitation which not only results in excess power consumption, but also require a relatively large chip area and extra power for instruction buses that are wider than necessary. By reference to Figure 3, internal instruction buses 370 and 380 in the present invention have a width no greater than 64 bits, to handle instruction packets that are 112 bits wide (such as exemplary instruction packets 410 and 430 in the present application). As stated in the present application:

“[A]ccording to the present embodiment of the invention, the width of each internal instruction bus 370 or 380 does not need to be greater than 64 bits in order to transport the various issue groups to thread A processing

unit 303 or thread B processing unit 305 for execution. However, according to conventional VLIW processors, an internal instruction bus having a width of at least 112 bits would be required. The reason is that, according to conventional VLIW processors, it is possible that all of the instructions in a VLIW packet belong to a single issue group. In other words, it is possible that the VLIW packet contains only one issue group. As such, all of the instructions contained in the VLIW packet must be transported simultaneously to a processing unit for execution. Thus, in the above examples, the conventional VLIW processor would need a 112-bit wide internal instruction bus. As is known in the art, power is consumed when each bus line corresponding to a particular bit is charged or discharged. Moreover, and in general, each line in the bus corresponding to a particular bit consumes some power in each clock cycle even when that particular bus line is not being used to transfer information during that clock cycle." See page 21, lines 1-15 of the present application.

Independent claims of the present invention specifically require a busing architecture with internal instruction buses no greater than 64 bits wide for transport of issue groups to each thread of the VLIW processor. In contrast, Tremblay is directed to a VLIW processor containing independent clustered functional units capable of parallel processing of instructions. More particularly, Tremblay is directed to a core processor 100, and media processing units 110 each disclosed as having an instruction cache 210, an instruction aligner 212, an instruction buffer 214, a pipeline control unit 226, a split register file 216, execution units, and a load/store unit 218. The media processing units 110 use execution units for executing instructions. The execution units include three media functional units (MFU) 220 and one general functional unit (GFU) 222. The media functional units 220 are disclosed to be multiple single-instruction-multiple-datapath (MSIMD) media functional units. Each of the media functional units 220 is disclosed as capable of processing parallel 16-bit components. Various parallel 16-bit

Attorney Docket No.: 00CON102P

operations supply the single-instruction-multiple-datapath capability for the processor 100 including add, multiply-add, shift, and compare. See, for example, Figure 3 of Tremblay and column 6, lines 51-67.

However, Tremblay does not disclose or even suggest a busing architecture for reducing the width of instruction buses, as disclosed and claimed by independent claims of the present invention. In other words, Tremblay does not disclose or suggest a busing architecture with internal instruction buses no greater than 64 bits wide for transport of issue groups to each thread of the VLIW processor. As stated above, Tremblay is not directed to, nor does it even suggest, the forced limitation of each issue group to any number of bits or, more particularly, to 64 bits. Indeed, the portion of Tremblay relied upon by the Examiner states that each subinstruction can be between 16 and 64 bits long, while each issue group in the VLIW packet would consist of four to eight instructions, thus ranging between 64 and 512 bits: "Typical bit lengths of a subinstruction commonly range from 16 to 64 bits per functional unit to produce an instruction length often in a range from 64 to 512 bits for VLIW groups from four to eight subinstructions." Tremblay, column 2, lines 2-5. In other words, far from limiting the number of bits in each issue group to 64 bits, Tremblay indicates that the number of bits in each of its issue groups can be a wide range, starting from 64 bits (and up to 512 bits). However, the present invention is directed to a two-thread processor, with each thread being required to process an issue group less than or equal to 64 bits.

In addition to the above differences, the independent claims have been further amended to distinctly point out another difference between the invention and Tremblay. That is, Tremblay clearly discloses and in fact requires a sharing of a common memory, referred to as “Data Cache 106” in Figure 2 of Tremblay and also shown as “Shared Data Cache and Synchronization Area” which is coupled to media processing units 110 and 112 in Figure 3 of Tremblay. In this regard, Tremblay states that: “The data cache 106 allows *fast data sharing and eliminates the need for a complex, error-prone cache coherency protocol between the media processing units 110 and 112.*” Tremblay, column 5, lines 33-37 (emphasis added). Moreover, throughout the disclosure of Tremblay, it is made clear that both media processing units 110 and 112 obtain data and operands from the same shared data cache 106.

In contrast, the present invention is directed to at least two threads for processing issue groups of instructions, each issue group operating on data fetched from separate memory modules, communicating exclusively to a separate thread memory. As stated in the present application:

“Load/store units 314 and 316 perform memory fetches from thread A memory 304 and load the fetched data into load/store register file 334 and scalar register files 324 and 322 as well as vector register file 326. Data paths 306 and 308 typically contain a variety of different types of functional units such as multiply-accumulate (“MAC”) units, adders, subtractors, logical shifts, arithmetic shifts, and any other functional units for performing mathematical or logical operations. The result of operations performed by

Attorney Docket No.: 00CON102P

data paths 306 and 308 on scalar or vector operands provided by scalar register files 324 and 322 and vector register file 326 can be stored in thread A memory 304.” *Page 3, lines 9-16 of the present application.*

It is further stated in the present application that:

“Load/store units 318 and 320 perform memory fetches from thread B memory 302 and load the fetched data into load/store register file 336 and scalar register files 328 and 330 as well as vector register file 332. Data paths 310 and 312 typically contain a variety of different types of functional units such as multiply-accumulate (“MAC”) units, adders, subtractors, logical shifts, arithmetic shifts, and any other functional units for performing mathematical or logical operations. The result of operations performed by data paths 310 and 312 on scalar or vector operands provided by scalar register files 328 and 330 and vector register file 332 can be stored in thread B memory 302.” *Page 14, line 17 through page 15, line 2 of the present application.*

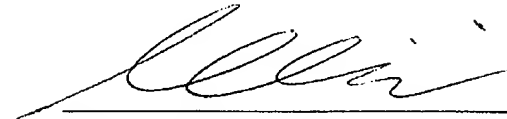
To distinctly claim this feature of the present invention, independent claim 1 has been amended to require: “each said one of said third plurality of issue groups performing an operation on data fetched from an exclusive thread memory communicating with only one of said first plurality of threads, a result of said operation being stored back in said exclusive memory thread communicating with said only one of said plurality of threads.” Independent claims 9 and 21 have been amended to require similar limitations. Thus, the invention’s disclosure and claims are directed to processing issue groups in two distinct threads, not even sharing a common cache or a common memory for fetching the raw

Attorney Docket No.: 00CON102P

operands. While Tremblay in fact discourages, or teaches away from, separate memory for holding data fetched by media processing units 110 and 112 by stating that: "The data cache 106 allows *fast data sharing and eliminates the need for a complex, error-prone cache coherency protocol between the media processing units 110 and 112.*" Tremblay, column 5, lines 33-37 (emphasis added). Thus, in addition to all other reasons stated above, the present invention is further patentably distinguishable over Tremblay, due to the claim amendments made in the present response.

For all the foregoing reasons, Applicants respectfully submit that the present invention, as defined by independent claims 1, 9, and 21 is not taught, disclosed, or suggested by the art of record. Thus, independent claims 1, 9, and 21 are patentably distinguishable over the art of record. As such, the claims depending from independent claims 1, 9, and 21 are, *a fortiori*, also patentable for at least the reasons presented above and also for additional limitations contained in each dependent claim. Thus, and for all the foregoing reasons, an early Notice of Allowance directed to claims 1-9, 11-15, and 21-28 remaining in the present application is respectfully requested.

Attorney Docket No.: 00CON102P

Respectfully Submitted,
FARJAMI & FARJAMI LLPDate: 10/10/05Michael Farjami, Esq.
Reg. No. 38,135FARJAMI & FARJAMI LLP
26522 La Alameda Ave., Suite 360
Mission Viejo, California 92691
Telephone: (949) 282-1000
Facsimile: (949) 282-1002CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this correspondence is being filed by facsimile transmission to United States Patent and Trademark Office at facsimile number 571-273-8300 on the date stated below. The facsimile transmission report indicated that the facsimile transmission was successful.

Date of Facsimile: 10/10/05Christina Carter
Name of Person Performing Facsimile TransmissionChristina Carter 10/10/05
Signature DateCERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed: Mail Stop RCE; Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

Date of Deposit: _____

Name of Person Mailing Paper and/or Fee_____
Signature Date